

AMENDMENTS TO THE CLAIMS

The following is a copy of Applicant's claims that identifies language being added with underlining ("____") and language being deleted with strikethrough ("~~---~~"), as is applicable:

1. (Previously Presented) A method of storing an XML document in a relational database comprising:

(a) parsing each node of the XML document into constituent parts, including parsing elements and, where an element has an attribute, the attribute of that element;

(b) associating a unique identifier with a respective parsed node of the document which identifies, absolutely, the hierarchical position of the node in the document; and

(c) storing each parsed constituent part of each node with its identifier in a table of a relational database.

2. (Original) A method according to claim 1, wherein the identifiers are associated such that a predetermined ordering of the identifiers and associated nodes in the database produces a predetermined ordering of nodes.

3. (Original) A method according to claim 2, wherein the predetermined ordering of the nodes is that produced by a depth first traversal of a tree representation of the hierarchical document.

4. (Previously Presented) A method according to claim 1, wherein the identifier includes a separate character position for each hierarchical level in the document which is traversed to reach the associated node in the hierarchical document.

5. (Original) A method according to claim 4, wherein a unique prefix character is used each time the number of nodes in a particular hierarchical level exceeds the unique characters in the identifier alphabet.

6. (Currently Amended) A method according to claim 1, wherein at least one database table entry includes a document identifier which identifies the hierarchical document from which an a node has been parsed.

7. (Previously Presented) A method according to claim 1 wherein at least one database table entry includes a value field which records a value of the node in the table entry.

8. (Previously Presented) A method according to claim 1 wherein at least one database table entry includes a type field which indicates a characteristic type of the node in the table entry from a predetermined set of types.

9. (Previously Presented) A method according to claim 1, wherein the hierarchical document is an XML document.

10. (Original) A method according to claim 9, wherein at least one database table entry includes a type field which indicates a characteristic type of the node in the table entry from a predetermined set of types and wherein the set of types includes text node, element node, attribute node and/or processing instruction.

11. (Original) A method according to claim 9 or claim 10, wherein the database table includes YPath and ZPath indexes pointing to predetermined respective entries in respective node and ZPath database tables.

12. (Currently Amended) A relational database comprising
a table having an a node field for storing each parsed constituent part of each node of an XML document including elements and, where an element has an attribute, the attribute of that element; and
an identifier field for storing an identifier associated with each respective node stored in the node field, wherein the identifier identifies, absolutely, the hierarchical position of the node in the document.

13. (Currently Amended) A database according to claim 12, wherein at least one database table entry includes a document identifier field for storing a document

identifier which identifies the hierarchical document from which an a node has been parsed.

14. (Currently Amended) A database according to claim 12 or claim 13, wherein at least one database table entry includes a value field for recording a value of an a node in the respective table entry.

15. (Currently Amended) A database according to any of claims 12 to 14, wherein at least one database table entry includes a type field for storing an indication of a characteristic type of an a node in the respective table entry from a predetermined set of types.

16. (Original) A database according to any of claims 12 to 15, wherein the database table includes node and ZPath indexes referencing respective entries in respective node and ZPath database tables in the database.

17. (Original) A database according to claim 16 wherein the YPath table includes fields for storing XPath element names and document IDs.

18. (Original) A database according to claim 16 or claim 17, wherein the ZPath table includes fields for storing XPath integer indexes and document IDs.

19. (Previously Presented) A method of writing an XML document comprising:

- (a) reading data from a relational database which is representative of constituent parts of each node of the XML document, the constituent parts comprising any elements of the node and, where an element has an attribute, the attribute of that element;
- (b) generating predetermined software events for respective read nodes; and
- (c) passing the software events to a content handler which is arranged to translate each software event into a written node of the XML document, each written node being associated with a unique identifier which identifies, absolutely, the hierarchical position of a respective written node in the document.

20. (Previously Presented) A computer readable medium carrying a program which when executed on a computer causes storing of an XML document in a relational database by:

- (a) parsing each node of the XML document into constituent parts, including parsing elements and, where an element has an attribute, the attribute of that element;
- (b) associating a unique identifier with a respective parsed node of the document which identifies, absolutely, the hierarchical position of the node in the document; and
- (c) storing each parsed constituent part of each node with its identifier in a table of a relational database.

21. (Previously Presented) A computer readable medium carrying a program which when executed on a computer causes storing of an XML document in a relational database by:

(a) receiving software events representing respective parsed nodes of the XML document;

(b) associating a unique identifier with the respective parsed nodes of the document which identifies, absolutely, the hierarchical position of the node in the document; and

(c) storing constituent parts of each node of the document with its identifier in a table of a relational database, the constituent parts comprising any elements of the node and, where an element has an attribute, the attribute of that element.

22. (Previously Presented) A computer readable medium carrying a program which when executed on a computer causing writing of an XML document by:

(a) reading data from a relational database which is representative of constituent parts of each node of the XML document, the constituent parts comprising any elements of the node and, where an element has an attribute, the attribute of that element;

(b) generating predetermined software events for respective read nodes; and

(c) passing the software events to a content handler which is arranged to translate each software event into a written node of the XML document, each written node being associated with a unique identifier which identifies, absolutely, the hierarchical position of a respective written node in the document.

23. (Previously Presented) The method of claim 1 further comprising:
reading data from the relational database which is representative of each node of the XML document; and

writing the data into a document such that the document contains each element and attribute of each node of the XML document at the appropriate hierarchical positioning as indicated by the unique identifier for each node.

24. (Previously Presented) The computer readable medium of claim 20, the program causing the computer to:

read data from the relational database which is representative of each node of the XML document; and

write the data into a document such that the document contains each element and attribute of each node of the XML document at the appropriate hierarchical positioning as indicated by the unique identifier for each node.

25. (Previously Presented) The computer readable medium of claim 21, the program causing the computer to:

read data from the relational database which is representative of each node of the XML document; and

write the data into a document such that the document contains each element and attribute of each node of the XML document at the appropriate hierarchical positioning as indicated by the unique identifier for each node.